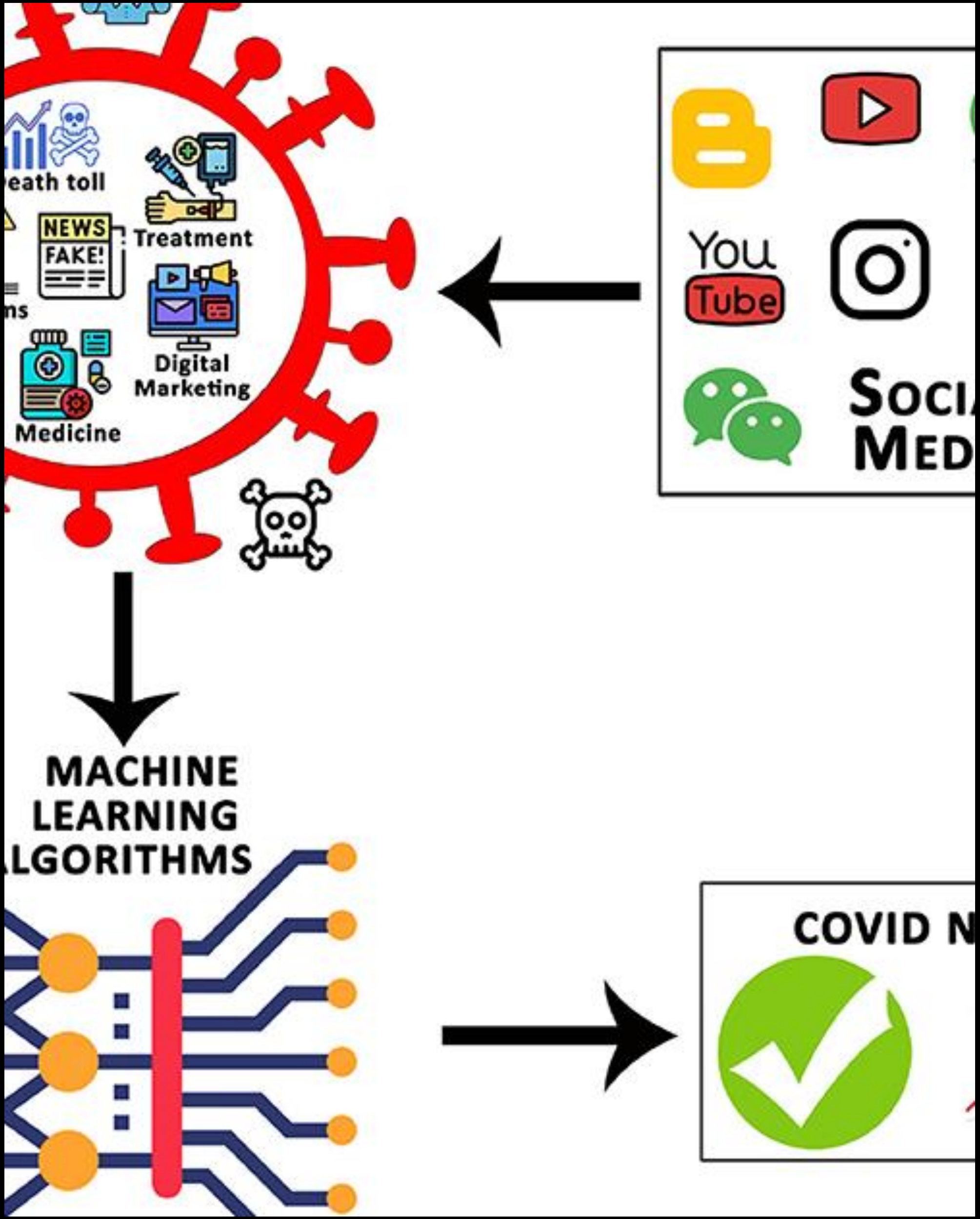
Fake News Detection with Machine Learning

An introduction to using machine learning techniques to detect fake news.

This project involves developing machine learning models to detect fake news articles on the internet. We will train classifiers on datasets of real and fake news stories to try to accurately flag fake content.



Problem Statement

False information spreads rapidly

Social media's algorithms can amplify fake news quickly without fact checking

Damages trust and causes harm

Fake news erodes public trust and can influence elections or incite violence

Difficult for users to identify

Fake news is intentionally misleading and can seem credible to readers

Fake news on social media is a serious problem that requires automated detection to stop its spread and influence.

Data Collection

Data Gathering

Data Cleaning

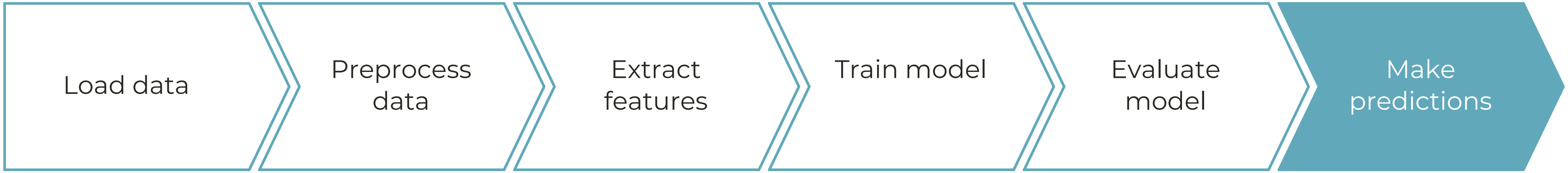
Feature Engineering

Collected a dataset of real and fake news articles from various sources.

Preprocessed the news articles by removing irrelevant information and formatting the data.

Extracted useful textual and linguistic features from the news articles that can help train the model.

Flowchart for Fake News Detection



Load data

Load training dataset containing true and fake news articles

Preprocess data

Tokenize text data, remove stopwords, etc.

Extract features

Convert text into numerical feature vectors like TF-IDF

Train model

Train a machine learning model like logistic regression on feature vectors and labels

Evaluate model

Test model performance on a held-out test set

Make predictions

Use trained model to classify new unseen articles as real or fake

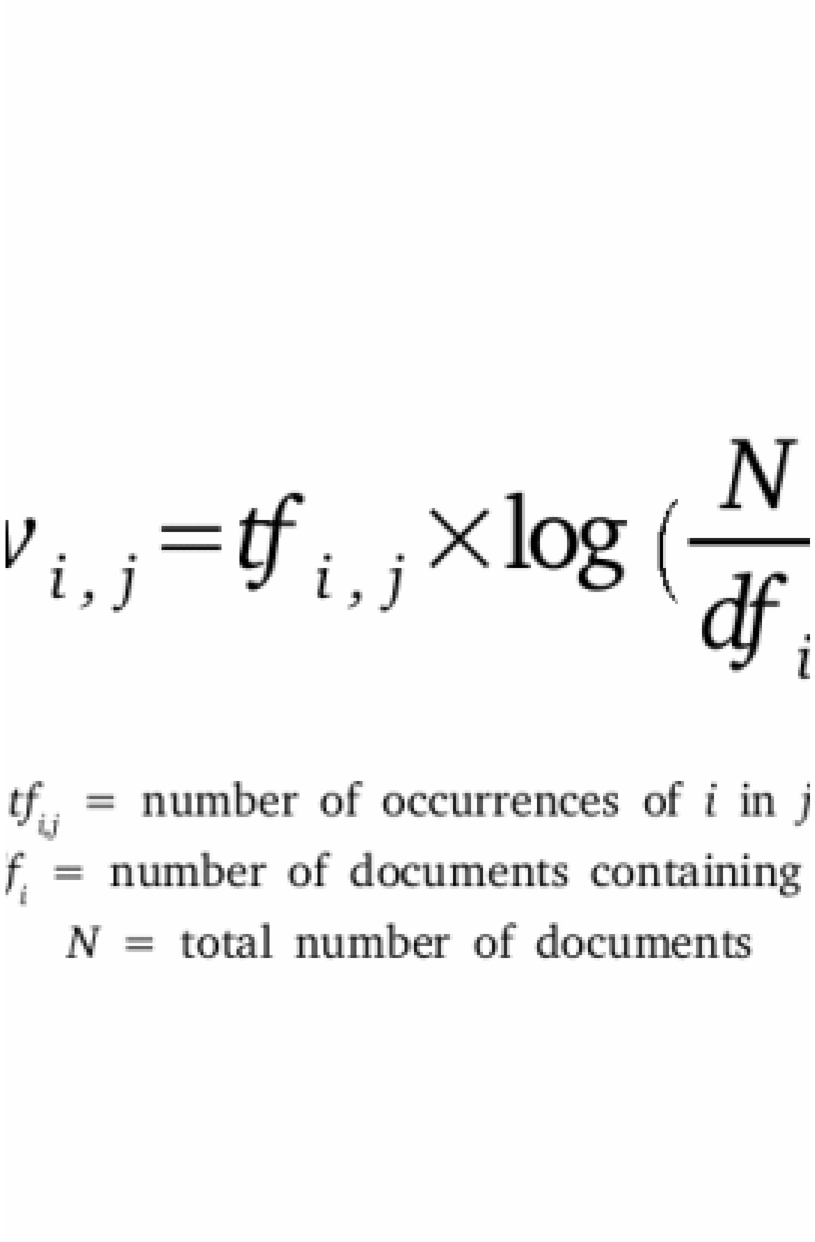
Tokenization with TF-IDF

tfidf, which stands for term frequency-inverse document frequency, is an algorithm used to tokenize text data when doing natural language processing tasks like text classification. It assigns a weight to words that appear frequently in a document relative to the rest of the corpus.

```
# Extract texts and labels
texts = dataset["text"]
labels = dataset["label"]

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(texts, labels, test_size=0.2,
random_state=42)

# Tokenization and TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_matrix_test = tfidf_vectorizer.transform(X_test)
```



$$v_{i,j} = tf_{i,j} \times \log \left(\frac{N}{df_i} \right.$$

$tf_{i,j}$ = number of occurrences of i in j

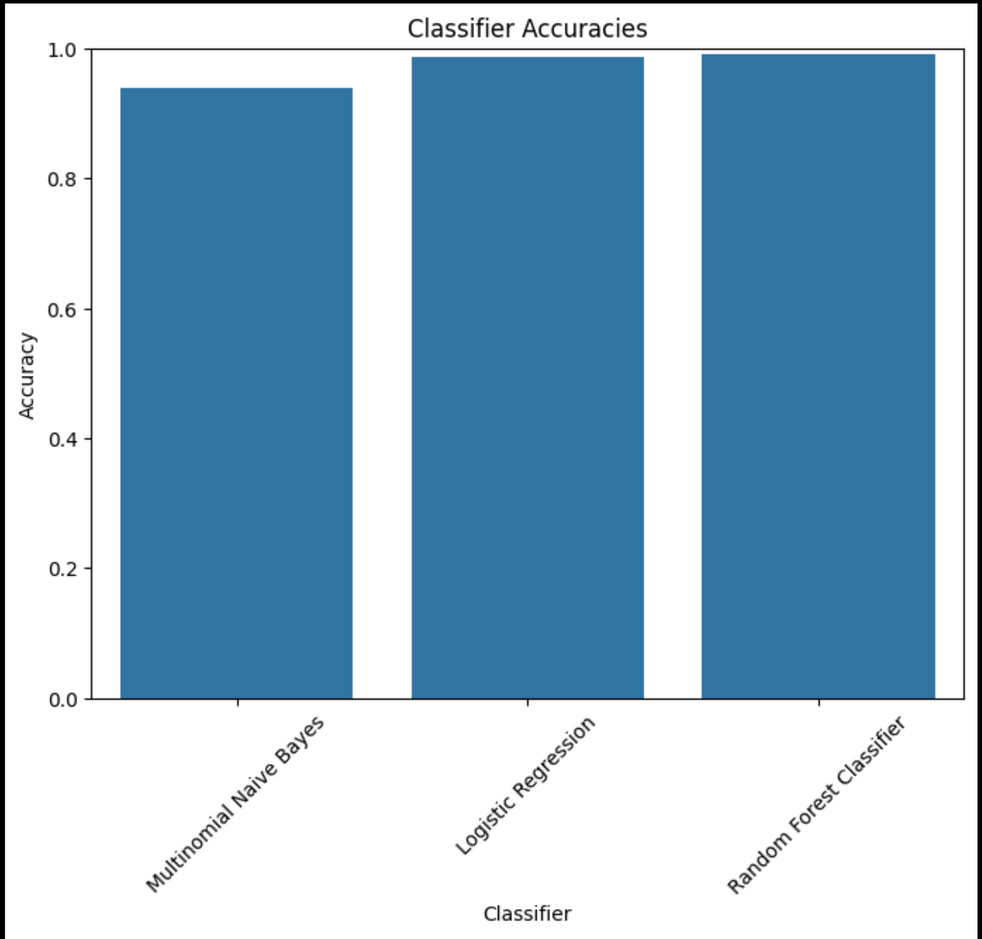
f_i = number of documents containing

N = total number of documents

Data Prediction

```
# Train classifiers
classifiers = {
    "Multinomial Naive Bayes": MultinomialNB(),
    "Logistic Regression": LogisticRegression(),
    "Random Forest Classifier":
RandomForestClassifier(n_estimators=100,
random_state=42)
}
accuracies = {}

for name, clf in classifiers.items():
    clf.fit(tfidf_matrix_train, y_train)
    y_pred = clf.predict(tfidf_matrix_test)
    accuracy = accuracy_score(y_test, y_pred)
    accuracies[name] = accuracy
    print(f"{name} Accuracy: {accuracy:.4f}")
```



Models Used

Multinomial Naive Bayes

A probabilistic classifier that uses Bayes theorem to calculate the probabilities of each class given the features.

Random Forest

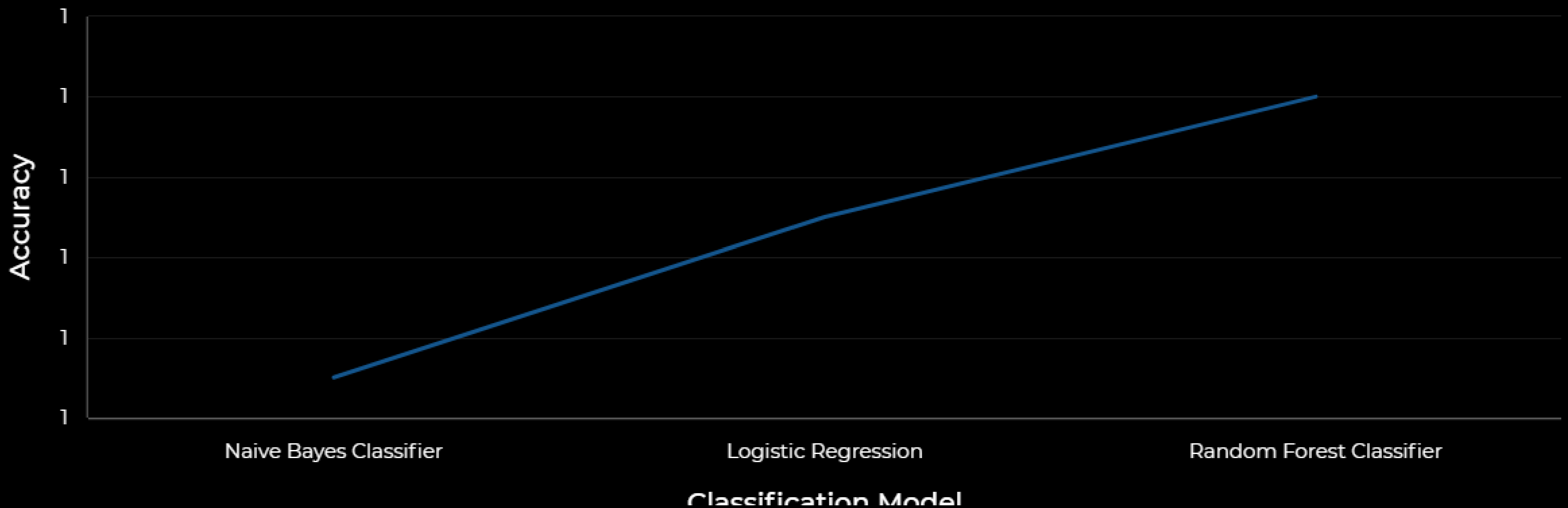
An ensemble method that constructs multiple decision trees and combines their predictions to improve overall performance.

Logistic Regression

A statistical model that estimates the probability of an event occurrence using a logistic function.

These 3 supervised machine learning models were used to classify news articles as real or fake based on their text content.

Model Accuracy Comparison

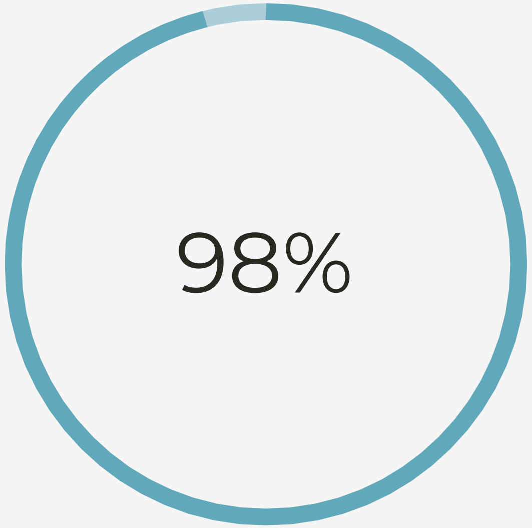


The Random Forest classifier performed the best accuracy on detecting fake news.

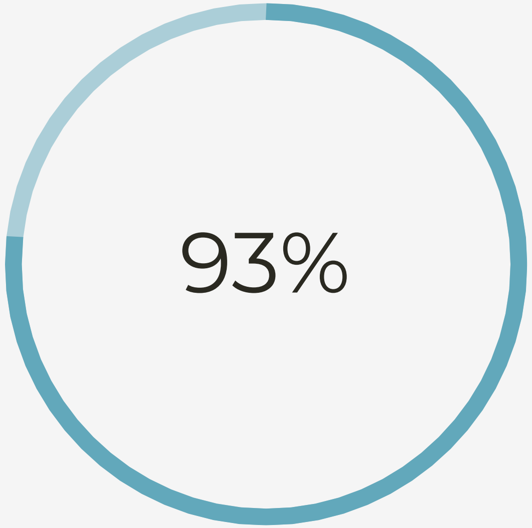
*Accuracy results from training multiple models on the fake news dataset.

Model Evaluation

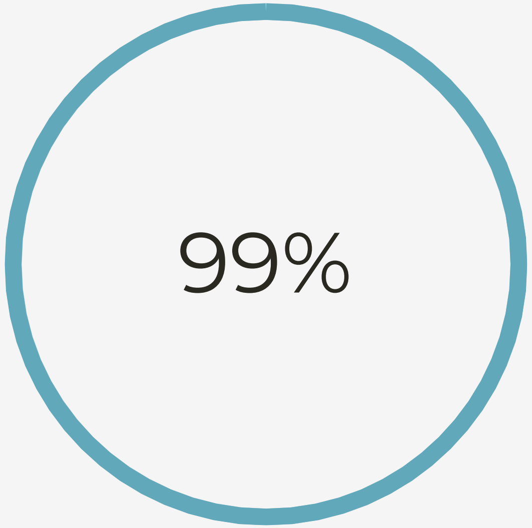
Accuracy scores in % for Logistic Regression, SVM, Naive Bayes and Random Forest models



Logistic Regression



Naive Bayes



Random Forest

Confusion Matrix

Confusion matrix shows performance of a classification model on test data

It's a table that compares predicted labels vs actual labels to quantify types of errors

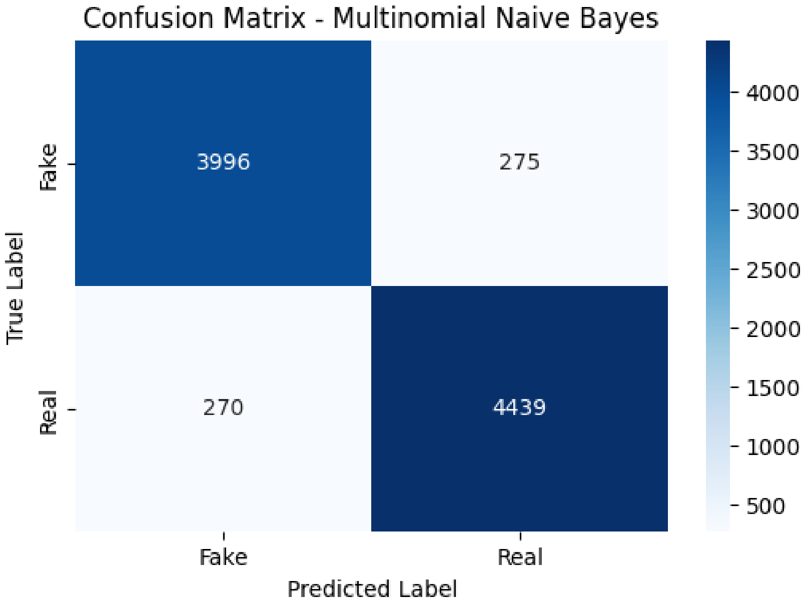
Helps identify bias of a model towards a class

Can see if a class is misclassified as another specific class more often

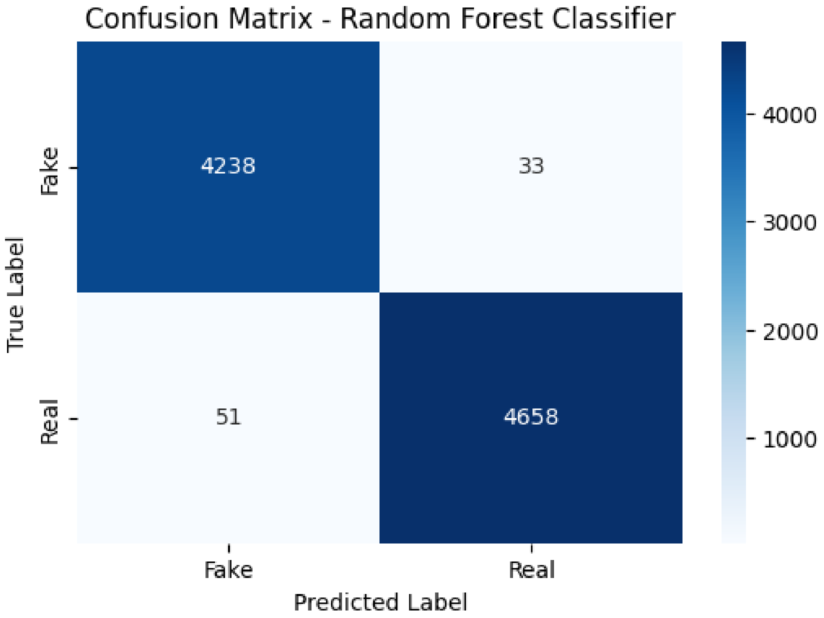
Quantifies accuracy, precision, recall for each class

Important metrics to choose right model for problem

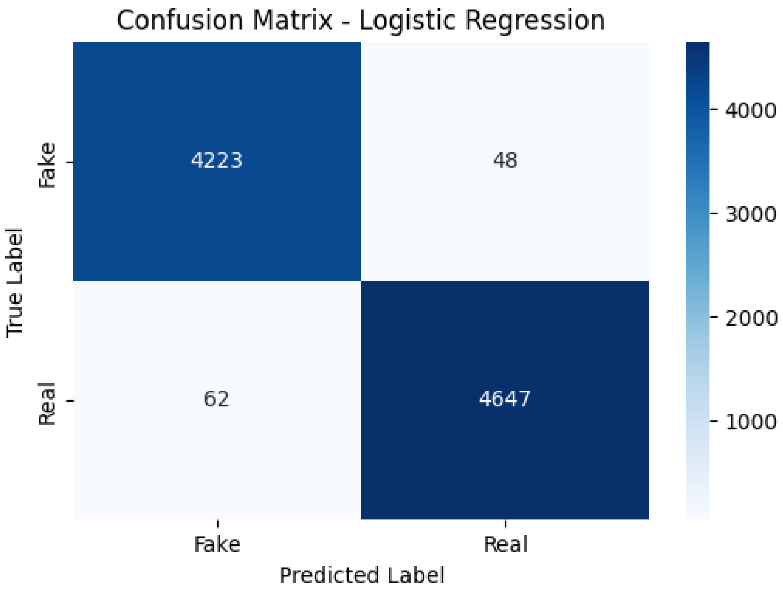
Confusion matrix provides insights into model performance for each class



Multinomial Naive Bayes



Random Forest



Logistic Regression

Next Steps

Collect more training data

Gather a larger dataset of news articles labeled as real or fake to improve model accuracy.

Try larger neural networks

Experiment with larger and more complex neural network architectures like LSTMs and Transformers.

Address class imbalance

Use techniques like oversampling or class weights to handle imbalanced classes in the training data.

With more data and larger networks, we can build a more accurate fake news detector.